University of Edinburgh

School of Law

Programme title

LL.M. in Innovation, Technology and the Law

Dissertation title

What does Privacy by Design mean for Digital Rights Management?

Examination number

4236100

Word count

9,941

Contents

1. Introduction

The chequered history of Digital Rights Management has left the technology with a significant public relations problem. Users have become distrustful of it, and for good reason. At its heart, however, Digital Rights Management (hereinafter "DRM") is simply a set of value-neutral technical approaches to protecting copyright, something most would not argue against. At some point, however, the application of the technology went beyond this core purpose, not just in protecting the rights of owners beyond what copyright law mandates,[1] but also by infringing the fundamental rights of users, most notoriously the right to privacy and to protection of personal data as enshrined by the European Convention on Human Rights[2] and the Charter of Fundamental Rights.[3]

Given this background and the damaged reputation of DRM, how can it be recast to meet the aims for which it was originally designed, whilst simultaneously upholding individuals' privacy rights in a digital era when fresh attacks on them come to light on almost a daily basis? And how can DRM fit with the requirement for privacy by design in the soon to be enacted General Data Protection Regulation?[4]

European institutions have highlighted a desire for market-driven approaches to privacy by design which can respond to the changing technological and economic landscape, while maintaining a base level of protection for European citizens' fundamental rights.[5] This paper suggests a novel approach that integrates software and legal process modelling in order to facilitate efficient, low-cost regulatory compliance whilst simultaneously upholding the user-oriented goals of privacy by

1 For more on the problem of DRM "over-reaching" see J.E. Cohen, 'A Right to Read Anonymously: A Closer Look at Copyright Management in Cyberspace' (1995) 28 *Conn. L. Rev.* 981.

2 Convention for the Protection of Human Rights and Fundamental Freedoms (1950) ("ECHR"), Art. 8. The focus in this paper is on European law, although of course privacy and data protection considerations reach far beyond European borders.

3 Charter of Fundamental Rights of the European Union (2000), Arts. 7 and 8.

4 Regulation on the protection of individuals with regard to the processing of personal data and on the free movement of such data (hereinafter the "draft GDPR"), Art. 23. For the most recent draft text see "General Data Protection Regulation – Preparation of a General Approach" (European Council 9565/15, Brussels, 11 June 2015) – subsequent references are to this latest text. The Regulation will come into force two years after the text is agreed (Art. 91), which is expected to happen by the end of 2015.

5 See European Data Protection Supervisor, 'EDPS Recommendations on the EU's Options for Data Protection Reform' (2015) Opinion 3/2015, p. 8 and Amendment 27 of the Opinion of the Committee on the Internal Market and Consumer Protection on an earlier draft of the GDPR (European Parliament A7-0402/2013).

design. The approach has the potential to bridge the gap between the legal and technical professions which at present poses a significant obstacle to practical implementations of privacy by design that are able to balance the interests both of enterprises and of users.

The paper begins by setting out the legal landscape of privacy and data protection law, as well as the privacy by design provisions in the draft GDPR. It then considers the primary DRM approaches and the ways in which they can be abused to undermine user privacy, before describing a notorious incident which demonstrates the significant social and commercial impact that can result from its abuse, and therefore why rightholders and DRM producers have an interest in adapting their systems to be regulatorily compliant. Thereafter, the discussion sets out the foundations of the novel approach and considers its suitability for privacy by design implementation under the draft GDPR regime, before concluding.

2. The legal context

2.1 Privacy and data protection law

Data protection as a concept can be traced back to the quest to assert and protect fundamental human rights in the aftermath of the second world war. In 1950 the right to privacy, from which data protection flows, was enshrined by Article 8 of the ECHR. In 1968, at the UN International Conference on Human Rights in Tehran, concerns were raised about the ways in which the fundamental right to privacy might be impacted by newly emerging technologies for recording information – this was the first discussion of data protection on the international stage.[6] The first domestic data protection legislation was enacted in the German state of Hesse in 1970, followed by Sweden in 1973.[7] In 1980, having identifying the potential economic value of personal data, the Council of the OECD published guidelines[8] aimed at striking a balance between the commercial interest in exploiting personal data on the one hand, and users' privacy rights on the other, noting that the latter had the potential to "cause serious disruption in important sectors of the economy".[9] The guidelines were

6 F.H. Cate, 'EU Data Protection Directive, Information Privacy, and the Public Interest, The' (1994) 80 *Iowa L. Rev.* 431 at p. 431; G.G. Fuster, *The Emergence of Personal Data Protection as a Fundamental Right of the EU* (Springer Science & Business, 2014), p. 40.

7 Cate (fn 6), p. 431.

8 'Guidelines on the Protection of Privacy and Transborder Flows of Personal Data' (OECD, 1980).

9 *Ibid*., preface.

the first to set out eight data protection principles of (i) collection limitation *(*the collection of personal data should be limited, fair and lawful, and should take place with the knowledge and consent of the data subject), (ii) data quality *(*data gathered should be relevant to the purpose(s) for which they will be used and should be accurate, complete and kept up-to-date), (iii) purpose specification (the purpose of the data collection should be specified, and further unspecified uses, or uses incompatible with the original purposes, are prohibited without notice), (iv) use limitation (changes to processing purposes require the consent of the data subject unless legally mandated), (v) security (personal data should be reasonably secured against loss and unauthorised access), (vi) openness (data controllers should be open about their identity and location as well as their practices and policies regarding personal data), (vii) individual participation (data subjects should be able to view, and have erased, rectified, completed or changed information held about them), and (viii) accountability (data controllers must comply with measures implementing the principles and are accountable for failures to do so).

These principles have been formulated in various ways across subsequent instruments, but their essence runs throughout. They appear in the 1981 Council of Europe Convention on personal data[10] and the 1995 DPD.[11] Since the DPD's enactment the principles have appeared in an ISO standard[12] and, of course, the draft GDPR.[13] At the core of each of these instruments is the concept of personal data, meaning data which identify (or are capable of identifying[14]) a natural person.[15] This crucial threshold can play an important part in implementing privacy by design, as we shall see below.

2.2 Privacy by design

The concept of privacy by design (hereinafter "PbD") dates from around the late

10 Council of Europe Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data (1981), Arts. 5-8.
11 DPD Arts. 6-8.
12 'ISO/IEC 29100:2011 - Information Technology - Security Techniques - Privacy Framework' (ISO/IEC, 2011) BS ISO/IEC 29100:2011.
13 Draft GDPR, Arts. 5-10.
14 In Europe, at least, under DPD Art. 2(a) (cf. the United States where identifiability is much less strongly regulated. For a discussion of this point see P.M. Schwartz and D.J. Solove, 'Reconciling Personal Information in the United States and European Union' (2014) 102 *California Law Review* 877.
15 1981 Convention Art. 2(a); DPD Art. 2(a); draft GDPR Art. 4(1); 'ISO/IEC 29100:2011 - Information Technology - Security Techniques - Privacy Framework' (fn 12), para. 2.8.

1990s. It first gained prominence in Canada through the work of Ontario's former Information and Privacy Commissioner, Ann Cavoukian.[16] The first mention in official European literature appears in the Commission's 2010 Communication entitled “A Digital Agenda for Europe”, which states tersely, and only in a footnote, that

> “[PbD] means that privacy and data protection are embedded throughout the entire life cycle of technologies, from the early design stage to their deployment, use and ultimate disposal.”[17]

This definition was repeated verbatim, again in a footnote, in another 2010 Commission Communication entitled “A comprehensive approach to personal data protection in the European Union”.[18] Two years later, the concept of PbD was more fully articulated in Article 23 of the draft GDPR, where it is referred to as “data protection by design”.[19] Having gone through significant amendment in the European Parliament and Commission readings, Article 23(1) currently reads

> “Having regard to available technology and the cost of implementation and taking account of the nature, scope, context and purposes of the processing as well as the likelihood and severity of the risk for rights and freedoms of individuals posed by the processing, the controllers shall implement technical and organisational measures appropriate to the processing activity being carried out and its objectives, such as data minimisation and pseudonymisation, in such a way that the processing will meet the requirements of this Regulation and protect the rights of data subjects.”

Recital 61 uses similar language, suggesting that PbD measures “could” involve (inter alia) minimisation of processing, pseudonymisation and transparency. It states also that

---

16 A. Cavoukian, ‘Privacy by Design: Origins, Meaning, and Prospects for Assuring Privacy and Trust in the Information Era’ in George OM Yee (ed) at *Privacy Protection Measures and Technologies in Business Organizations: Aspects and Standards: Aspects and Standards* (IGI Global, 2012).

17 European Commission, ‘A Digital Agenda for Europe’ (2010) COM/2010/0245 final, note 21.

18 European Commission, ‘A Comprehensive Approach on Personal Data Protection in the European Union’ (2010) COM(2010) 609 final, note 30.

19 The two terms are treated as synonymous in the remained of this paper; for brevity the acronym “PbD” be used.

> "...producers of the products, services and applications [which process personal data] should be encouraged to take into account the right to data protection when developing and designing [them] and, with due regard to the state of the art, to make sure that controllers and processors are able to fulfil their data protection obligations."

The mention of pseudonymisation and data minimisation in Article 23(1) is welcome, and provides a suggestion on where to begin with technical implementations of PbD which was absent in the previous drafts.[20] Nevertheless there is little further guidance on how to achieve compliance under these provisions, leading to criticisms that they are vague,[21] that they "do not address technology producers [or] allow real technology design",[22] and that they provide data controllers with "little clue on how they should go about 'designing in' privacy".[23]

For the founding principles of PbD to be vague – Gürses et al note, for example, that some of Cavoukian's original principles are recursive, that is their definition repeats the term, as in "Privacy by Design is embedded into the design"[24] – is perhaps understandable given the new ground they were breaking well over a decade ago. In the present day, however, and especially in a Europe-wide legal instrument that is binding verbatim and has significant implications for worldwide information flows, such uncertainties are difficult to excuse.

As mentioned in the introduction, there is a desire to see the market-driven creation of PbD solutions which can respond quickly to changes in technology and social mores.[25] But without practical guidance on what technical approaches will be legally-

---

20 See for example Article 29 Data Protection Working Party, 'Opinion 01/2012 on the Data Protection Reform Proposals' (2012) Opinion WP 191, p. 11 <http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp191_en.pdf> accessed 4 July 2015.

21 S. Gürses, C. Troncoso and C. Diaz, 'Engineering Privacy by Design' (2011) 14 *Computers, Privacy & Data Protection* <https://www.cosic.esat.kuleuven.be/publications/article-1542.pdf> accessed 5 July 2015.

22 M. Pocs, 'Will the European Commission Be Able to Standardise Legal Technology Design without a Legal Method?' (2012) 28 *Computer Law & Security Review* 641 at p. 644.

23 B.-J. Koops and R. Leenes, 'Privacy Regulation Cannot Be Hardcoded. A Critical Comment on the "privacy by Design" Provision in Data-Protection Law' (2014) 28 *International Review of Law, Computers & Technology* 159 at p. 162.

24 A. Cavoukian and others, 'Privacy by Design: The 7 Foundational Principles', Principle 3 <https://www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf> accessed 16 May 2015.

25 European Data Protection Supervisor (fn 5), p. 8.

compliant the market is likely to respond by travelling the path of least resistance, which is likely to undermine users' and, ultimately, enterprise interests.

2.2.1 Risks arising from failure properly to implement PbD

The ambiguity surrounding PbD is problematic, particularly given the GDPR will have direct effect. From an economic perspective, innovation is chilled as only those enterprises with sufficient legal and financial resources to defend their actions will have the confidence to develop new products and services in areas where data protection and PbD law operate. The smaller, dynamic innovators who are often the source of important technological breakthroughs are likely to be discouraged by the prospect of legal action, financial penalties and adverse publicity that might arise should they misinterpret the ill-defined PbD provisions or inadvertently fail to implement them in their designs.[26] Alternatively, they may decide to take the risk and ignore the requirement for PbD, especially in the initial stages when its meaning is fluid and under interpretation (a precedent for this is the infamous EU "Cookie law",[27] which has been significantly defanged by domestic data protection authorities owing to both its ambiguity and impracticality[28]). Such an eventuality may be commercially attractive because the need to observe regulatory requirements is reduced, but it would make a mockery of Article 23 by undermining the very rights it is intended to protect.

Bearing in mind the conflicts inherent in a market where the enterprises creating the software that will be required to embody PbD goals are often the same enterprises which stand to gain from gathering personal data, there is a potentially concomitant effect arising from the centralisation of PbD innovation around those companies. In the absence of real competition between PbD approaches, and with the potential for a cartel-like closing of ranks amongst the embedded players, we may be left with token gestures towards PbD rather than concrete implementations that are demonstrably effective in upholding users' rights.[29]

---

26 The draft GDPR proposes penalties of up to €1m, or 2% of worldwide turnover, for breaches of the Art. 23 PbD requirements. See Art. 79(6).

27 Directive Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector (the ePrivacy Directive) 2002/58/EC 2002, Art. 5.

28 The UK Information Commissioner's Office, for example, now assesses only the top 200 websites in that jurisdiction, and even then only quarterly. See <https://ico.org.uk/action-weve-taken/cookies/> (accessed 15 August 2015).

29 Gürses, Troncoso and Diaz (fn 22), part 4.2.

Further below I discuss what the overarching concerns are of those aiming to implement privacy-friendly systems. The difficulty of PbD's ill-defined characteristics and the lack of concrete guidance can to an extent be sidestepped by shifting our focus away from *post hoc* compliance with new and untested regulatory ideas towards the application of what are well-established data protection values at a stage early enough in the design process that those values are, de facto, “designed in”. This is the basis of the novel approach presented by this paper. Before moving onto that discussion, I will first set out the landscape of DRM technology and how it can and has been used to undermine privacy.

3. DRM: a much-maligned technology

DRM refers to a range of technological approaches designed to protect rightholders' interests with regard to the unlicensed copying and distribution of their licensed works.[30] To that extent, DRM technologies are a perfectly legitimate and rational response to behaviour which threatens those interests. That behaviour has been radically enabled by the move towards mass digitisation of media content, where the creation of identical copies is trivial, fast and virtually costless. In spite of this, certain DRM implementations have gone beyond protecting rightholders' interests, extending the core copyright reservations beyond their normative scope and breaching user privacy in the process.

3.1 Approaches used in DRM systems

The aims of DRM are to control access to licensed works, prevent or limit unauthorised use of those works, identify the works, the rightholder and the authorised licensee, and to protect that information from tampering, corruption or the possibility of forging.[31] Individual DRM systems may prioritise and fulfil these aims differently according to commercial imperative, the medium on which they are intended to be used, limitations in the state-of-the-art or a lack of expertise/willingness on the part of the DRM vendor.

30 See Directive 2001/29/EC on the harmonisation of certain aspects of copyright and related rights in the information society, Recitals 25 and 28 and Arts. 2–4.

31 L.A. Bygrave, ‘Digital Rights Management and Privacy – Legal Aspects in the European Union’ in Eberhard Becker, Willms Buhse and Dirk Günnewig (eds) at *Digital Rights Management - Technological, Economic, Legal and Political Aspects* (Springer, 2003), p. 420; on forgeability, see J.A. Halderman and E.W. Felten, ‘Lessons from the Sony CD DRM Episode.’ [2006] *15th USENIX Security Symposium* 77, pp. 9–10.

There are four primary DRM strategies rightholders can employ to protect content, although they are not mutually exclusive – a given DRM system can employ more than one simultaneously.[32] Classified broadly from the least to the most privacy-invasive, the approaches are as follows:

3.1.1 Encryption and metadata (file-side DRM)[33]

This approach involves encrypting the contents of the file so that only approved software can decrypt it. The file may also include metadata that defines the ways in which a particular licensee is authorised to use the file – in such cases the DRM is said to control both access and use.[34] This approach is portable – the DRM in its entirety travels with the file and operates in a fairly passive manner such that if the holder of the file has the requisite software and/or decryption key then she will be able to access the file regardless of how she obtained them. In terms of privacy, this basic form of DRM is fairly benign – it is not designed to check who a given user is and *a fortiori* provides no mechanism for identification or behavioural tracking.

3.1.2 Associating files with the device or user (relationship-based DRM)

The "node locking" approach associates the file with a particular device, and involves generating a unique fingerprint from that device. The DRM algorithm can prevent access if it finds that the fingerprint it generates differs from the one embedded in the file. This approach has implications for privacy – upon purchase the user will be identified via the association with their device, and any subsequent re-association (for example if they wish to move the file to an upgraded device) will re-identify the user to the rightholder and confirm that she is still interested in that particular content.[35]

A more invasive approach involves identification of the user by authenticating them on whichever device they are using. The DRM relationship is thus formed between the rightholder and the individual, instead of between the rightholder and a particular

32 These classifications are borrowed in part from J. Feigenbaum and others, 'Privacy Engineering for Digital Rights Management Systems' in Tomas Sander (ed) at *Security and Privacy in Digital Rights Management* (Springer Berlin Heidelberg, 2002), pp. 79–81 <http://link.springer.com/chapter/10.1007/3-540-47870-1_6> accessed 16 May 2015.

33 Sometimes also referred to as the "containment" approach. See C. Woodford, 'Trusted Computing Or Big Brother - Putting the Rights Back in Digital Rights Management' (2004) 75 *University of Colorado Law Review* 253, p. 274.

34 *Ibid*.

35 Feigenbaum and others (fn 34), p. 80.

inanimate device which might be used by many people, or indeed no-one. The scope for privacy invasion is accordingly much wider as real-time authentication by definition identifies the individual, as well as providing behavioural metadata on matters such as frequency and timing of use which can be aggregated to generate new forms of data about the user that she will in most cases be wholly unaware of.[36]

3.1.3 Embedding restrictions in hardware (device-side DRM)

This is a more sophisticated version of relationship-based DRM, and involves embedding deep within the hardware itself a framework of restrictions that prevents the use of files in the absence of some external authorisation. In contrast with software-based DRM, these systems are built into not just core components of the operating system but also the physical fabric of the machine itself, making circumvention significantly more difficult[37] but also forcing users to accede to unprecedented levels of use authorisation checking and control over their machines.[38]

3.1.4 Providing access on-the-fly (server-side DRM)

The final DRM approach is fast becoming the norm[39] as Internet connectivity speeds improve to a point where the huge amounts of data storage required for high-quality media can be transferred on-the-fly ("streamed") to users with little or no waiting times. Server-side DRM employs a combination of some the above methods – the user will be authenticated with the service, her profile will be checked to identify what media she is authorised to access, and the data stream itself will be encrypted during transfer and decryptable only by proprietary software on the user's device.

In terms of privacy this is perhaps the most invasive form of DRM. In such systems the locus of control is very much within the server(s) providing the content, rather than the user's own machine.[40] Access is entirely contingent on the rightholder

36 Cohen (fn 1), p. 7; P. Vora, D. Reynolds and I. Dickinson, 'Privacy and Digital Rights Management' at *A position paper for the W3C workshop on Digital Rights Management* (Publishing Systems and Solutions Lab, Hewlett-Packard Laboratories, 2001), pp. 2–3 <http://www.w3.org/2000/12/drm-ws/pp/hp-poorvi2.html> accessed 16 May 2015.

37 Woodford (fn 35), pp. 257 and 281.

38 R. Anderson, 'Trusted Computing FAQ v1.1' (August 2003) <http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html> accessed 15 June 2015 Question 2.

39 As recently as 2002 this approach was described as "radical"- see Feigenbaum and others (fn 34), p. 80.

40 cf. Spiekermann and Cranor's suggestion that personal data processing should occur client-

accepting the user's credentials. Beyond this, the chain of events required to access protected content provides an unprecedented opportunity for the rightholder to undermine user privacy through the gathering of detailed data about the user's tastes and behavioural patterns.

3.2 The Sony BMG rootkit incident

This section discusses a particularly infamous incident which demonstrates the commercial consequences of misjudging how to deploy the technology. In the Autumn of 2005 Sony BMG, then the world's second largest record company,[41] included on several of its CD releases "rootkit"[42] software designed to prevent playback and "ripping"[43] of the music contained thereon by unauthorised (in the technical, rather than normative, sense) applications on users' Windows PCs. The software installed itself surreptitiously on users' machines and required the use of a proprietary media application to listen to the CD, to convert its tracks to digital files (which had to be in a DRM-protected format), or to create up to three copies of the disc.[44] This application would also check whether the computer was running other CD-ripping software and would cease playback and eject the CD if such software was detected.[45] As with many other forms of DRM these measures appeared superficially reasonable insofar as they were aimed at thwarting copyright infringement, however it soon became apparent that the software's behaviour had serious implications for user privacy.

After investigating a tip-off from a commenter on his blog post about the Sony BMG DRM software, security researcher Mark Russinovich discovered that the media player software was connecting to Sony BMG's servers to check for updated album artwork, and in the process sending an ID of the CD being played along with the

---

side as far as possible, in order to avoid undermining privacy. See S. Spiekermann and L.F. Cranor, 'Engineering Privacy' (2009) 35 *IEEE Transactions on Software Engineering* 67 (discussed further below).

41 T.L. McPhail, *Global Communication: Theories, Stakeholders, and Trends* (John Wiley & Sons, 2009), p. 131.

42 The term "rootkit" refers to characteristics of the software which seriously undermined the security of the systems it was installed on. For a fuller description see Halderman and Felten (fn 32).

43 "Ripping" refers to the process of converting a CD's audio tracks into digital music files to be stored on another medium.

44 Halderman and Felten (fn 33), p. 14.

45 *Ibid*, p. 6.

user's IP address.[46] In a comment on Russinovich's blog a user purporting to represent the company which developed the software suggested that this "phoning home" behaviour was designed purely to update an advertising banner in the media player software and that "no information is ever fed back or collected about the consumer or their activities".[47] This was disingenuous insofar as merely connecting to Sony BMG's servers, even for the innocent purpose of updating the advertising banner, *necessarily* fed information back: absent the user's employment of some technical measure to mask it, their IP address will always be made known to the server they are connecting to. Furthermore, the inclusion of the CD's identifier and metadata storing the time of the request meant that, however minimal it might be argued the communication was, and whatever it might (or might not) be used for, it was nevertheless data which Sony BMG was being sent by the DRM software. This, coupled with the lack of disclosure in the software's terms and conditions and the absence of a viable uninstaller for the software meant it constituted what is known as "spyware"[48] – that is, "[s]oftware that surreptitiously gathers information and transmits it to interested parties".[49]

3.2.1 Commercial and public relations impact

The fall-out from the Sony BMG rootkit incident was significant. Direct economic impacts came from a fall-off in the number of affected CDs being sold, the recall of millions of affected discs (reportedly $6.5m to recall 4.7m[50] of a total of around 25m CDs[51]), settlement payouts resulting from numerous lawsuits,[52] and of course resulting loss of sales and the cost of a DRM system which could no longer be used. From a public relations perspective the company's reputation was battered, with users threatening to boycott both Sony BMG music and its parent company's range

46 Mark Russinovich, 'More on Sony: Dangerous Decloaking Patch, EULAs and Phoning Home' <http://blogs.technet.com/b/markrussinovich/archive/2005/11/04/more-on-sony-dangerous-decloaking-patch-eulas-and-phoning-home.aspx> accessed 8 June 2015.

47 Available on the Internet Archive at http://web.archive.org/web/20051124225410/http://www.sysinternals.com/blog/2005/11/more-on-sony-dangerous-decloaking.html#113115114916278358 (accessed 12 June 2015).

48 Halderman and Felten (fn 33), p. 14.

49 'Viruses, Spyware, and Malware' (*MIT Information Systems & Technology*) <https://ist.mit.edu/security/malware> accessed 2 August 2015.

50 'Sony BMG Recalls Copy Protected CDs' (*Billboard*, 18 November 2005) <http://www.billboard.com/articles/news/60609/sony-bmg-recalls-copy-protected-cds> accessed 2 August 2015.

51 D.K. Mulligan and A. Perzanowski, 'The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident' (2007) 22 *Berkeley Technology Law Journal* 1157, p. 1170.

52 *In re SONY BMG CD Technologies Litigation settlement agreement* [2005] New York Southern District Court No 1:05-CV-09575,.

of electronics[53] and commentators poking fun at Sony BMG's mistake. As a result the company became “inextricably associated with its misguided attempts at content protection”.[54]

Of course, the particular point in history when the rootkit incident took place perhaps served to maximise its impact. Broadband infrastructure was becoming ubiquitous, and with it the notional threat of file sharing to rightholders' revenue streams. DRM technology became sought after as a solution to this threat, and record labels and their DRM software suppliers were under pressure to stem the perceived tidal wave of piracy by whatever means necessary, sometimes (wittingly or unwittingly) cutting corners in the process.[55] Added to this, the costs inherent in the manufacture, distribution and recall of physical media will have had added to the impact. Despite this, the incident demonstrates the fragility of an entrenched position (Sony BMG was the world's second largest record label at the time of the incident[56]) in a market based on goodwill, where users' awareness of regulatory issues around privacy and data protection is increasing.[57] Indeed, the effects were not unique to Sony BMG. Privacy breaches have been shown to have a negative effect on stock market valuation,[58] while regulatory fines have reached into the hundreds of thousands[59] and financial settlements into the tens of millions.[60]

3.2.2 Legal analysis

From a European legal perspective the IP addresses collected by Sony BMG's DRM software directly are directly identifying and thus personal data,[61] bringing them

53 Mulligan and Perzanowski (fn 53), p. 1171.
54 *Ibid*.
55 Halderman and Felten (fn 33), pp. 2–3.
56 Mulligan and Perzanowski (fn 53), p. 1158.
57 See for example the European Commission's March 2015 survey of 28,000 EU citizens: ‘Data Protection Eurobarometer’ (2015) <http://ec.europa.eu/justice/newsroom/data-protection/news/240615_en.htm> accessed 31 July 2015.
58 A. Acquisti, A. Friedman and R. Telang, ‘Is There a Cost to Privacy Breaches? An Event Study’ [2006] *ICIS 2006 Proceedings* 94.
59 See for example the UK Information Commissioner’s Office, ‘Enforcement’ (7 July 2015) <https://ico.org.uk/action-weve-taken/enforcement/> accessed 2 August 2015.. Under the GDPR the upper ceiling for monetary penalties is set to increase significantly; see Art. 79.
60 See for example Federal Communications Commission, ‘AT&T To Pay $25M To Settle Investigation Into Three Data Breaches’ (8 April 2015) <https://www.fcc.gov/document/att-pay-25m-settle-investigation-three-data-breaches-0> accessed 2 August 2015.
61 See *Scarlet Extended SA v SABAM* [2011] EUECJ C-70/10, para. 51; Article 29 Data Protection Working Party, ‘Opinion 1/2008 on Data Protection Issues Related to Search Engines’ (2008) WP 148, p. 8.

within the regulatory auspices of the DPD[62] and, prospectively, the draft GDPR.[63] The failure to inform their customers of the software's behaviour arguably fell foul of the data protection principles' requirements of fairness and transparency, which are concerned with disclosure of the nature and extent of processing activities[64] – under the DPD data subjects must be “in a position to learn of the existence of the processing” and, more stringently under the draft GDPR, “informed of the existence of the processing operation and its purposes”. The processing purposes were neither specified nor explicit[65] – the license agreement made no mention of the software connecting to Sony BMG's servers, and in fact actively denied that it collected any personal data whatsoever.[66]

Furthermore, the data collected were excessive in relation to facilitating the DRM mechanism[67] which, as an example of simple file-side DRM (described above), did not require the collection of personal data at all in order to operate. Setting aside the other problems their system had as a mechanism of protecting their rights, the easiest way for Sony BMG to have avoided any data protection implications would have been to keep the underlying DRM mechanism requiring playback on their proprietary software but to eschew the functionality that communicated with their servers. If it was felt that this would be too great a reduction in functionality (that is, the banner ads and up-to-date album artwork were deemed to important a part of the product's selling point for them to be removed) then some other privacy enhancing mechanism could be used to remove the system from the data protection regime, for example anonymisation of users' IP addresses.

Perhaps the legal and technical options open to Sony BMG were unclear, or they knew about them but felt the potential commercial dividends were worth the risk. The novel approach discussed below aims to assist in clarifying such uncertainties, in order that enterprises developing DRM systems can do so with confidence that their

62 Recital 26 and Art. 2(a).
63 Recitals 23 and 24; Art. 4(2). Note the caveat in Recital 24, however, that identifiers such as IP addresses may not in every case constitute personal data.
64 DPD Art. 6(1)(a) and Recital 38, and draft GDPR Art. 5(1)(a) and Recitals 30 and 48, respectively.
65 DPD Art. 6(1)(b); draft GDPR Art. 5(1)(b).
66 A copy of the license agreement is available in the web archive at <http://web.archive.org/web/20051124012205/http://www.sysinternals.com/blog/sony-eula.htm> (accessed 13 June 2015).
67 DPD Art. 6(1)(c); draft GDPR Art. 5(1)(c).

software designs are legally-compliant. First I discuss the general range of concerns which at present face enterprises who wish to design privacy-compliant software.

4. Perspectives on privacy compliance

The over-arching range of factors involved in designing privacy compliance into a software system is complex. One might first think of consider the human-technical spectrum, ranging from the "fuzzy" regulation of natural language-based policies and agreements to the concreteness of regulation-by-code.[68] Then one might consider where on the spectrum of sophistication any given technical mechanism lies. Finally, and importantly for PbD, there is what might be termed the "design-runtime spectrum", which asks at what point in the development and release cycle of the software system the regulatory mechanism is implemented – nearer the beginning (design stage), or attached to the end product (runtime stage).

4.1 The human–technical spectrum

The two strands of Article 23 of the draft GDPR refer to the use of both technical and organisational (human) measures. Koops and Lennes are wary of too great an emphasis being placed on the former and consider that, due to the problems inherent in purely technology-based regulation, it will be organisational measures that are better placed to meet the aims of PbD.[69] Spiekermann and Cranor usefully re-frame the two ends of this spectrum as "privacy-by-policy" and "privacy-by-architecture",[70] whereby the former uses the traditional measures of privacy policies and other basic notice and consent mechanisms to achieve notionally the aims of the legislation – essentially, this represents the status quo. At the latter end the focus is on designing software to gather less personal data, a goal which is empowered by the approach suggested later in this paper.

At present businesses favour the primarily organisational, privacy-by-policy approach, because it can be applied easily to existing practises without impacting heavily on core business activities (by, for example, adding privacy policies or simple consent mechanisms). While this may have been sufficient in past years, it is now sub-optimal for at least three reasons. Firstly, the temptation will be to undermine

68 See L. Lessig, *Code: Version 2.0* (Basic Books, 2006).
69 Koops and Leenes (fn 23).
70 Spiekermann and Cranor (fn 41).

user privacy as technical systems remain “black boxes” from the user's perspective and because of the inherent “fuzziness” of regulations, policies and contracts expressed in natural language. Secondly, and as discussed above, as users become more aware of privacy concerns in the online environment they will rightly expect more explicit assurances as to how they are protected – anything less is likely to result in a chilling effect on users' economic activity.[71] Thirdly, businesses which focus solely on privacy-by-policy at the expense of privacy-by-architecture divert limited development resources away from development of the core product onto designing and implementing policies and consent measures. The content of those policies might diverge from the actual technical behaviour of the system and/or the nature of the consent may be insufficient or misinformed, inviting the risk of litigation or censure from regulatory authorities if such deficiencies are found to exist.

Even if we accept the concerns about regulatory tools which rely solely on technical measures, those concerns do not mean we cannot be *assisted* by computerised tools. The question is where the optimum point on the policy–architecture spectrum lies, and crucially, how this will look and where it should be located within the design-runtime spectrum, discussed below.

4.2 The technical sophistication spectrum

The technical approaches to implementing privacy within a software system have differing levels of complexity, efficacy and expense. On the less sophisticated end of the spectrum we have the most generic PET, encryption. Its strength is that it is technologically mature and simple to implement, but the drawback is that it is blunt in its operation, with file/service access and the reciprocal provision of personal data being all-or-nothing. This prevents fine-grained control, militating against a balancing of the interests of the enterprise and the user.

At the opposite end of the spectrum there is the translation of regulatory norms into representations which are directly comprehensible by artificial intelligence. With this “hard-coding”,[72] the machine can act directly on a digital “concept” which is directly analogous to its real-world counterpart. The benefits can be readily appreciated:

71 D. Chaum, ‘Security Without Identification: Transaction Systems to Make Big Brother Obsolete’ (1985) 28 *Commun. ACM* 1030.

72 Koops and Leenes (fn 23).

computers can interpret and enforce regulatory norms directly and with near-perfection, without the need for costly and time-consuming legal processes or human interpretation. This is the apotheosis of Lessig's concept of code-as-law.[73] But rather than, as in his thesis, code representing merely another separate regulatory modality operating within the broader mix, hard-coding envisions it instead as *subsuming* the very substance of the law, combining the already formidable power of regulation-by-code with the normative, democratic regulatory force of physical-world law. The result is, at least in theory, ideal – democratically legitimate legal norms enforced by code that can regulate perfectly.[74]

While these benefits are notionally attractive, there are democratic concerns surrounding the use of autonomous computer agents in regulation. Unlike the orthodox, human-centred regulatory framework, the development and enforcement of regulatory norms becomes centralised in the creator of the technology, sidestepping the separation of powers which traditionally acts as a check on the abuse of power.[75] Simultaneously, the perfection of code-based regulation encourages inflexibility and potentially authoritarianism as the role of humans in the process of assessing, adapting and enforcing regulatory norms is usurped.[76] A reliance by enterprise on the "ambient regulation"[77] that hard-coding would introduce would have the potential to discourage proper engagement and consideration of the regulatory environment within which their products operate and the rights and values which they have a bearing upon. This would likely result in commercially damaging incidents like the Sony BMG case and the undermining of users' fundamental rights.

Note that hard-coding should not to be confused with less complex techniques which aim to formalise the law into ontologies of legal artefact and relationship definitions whose predefined rules a computer can apply.[78] They lie closer to the middle of the

73 See generally Lessig (fn 69). See also J.R. Reidenberg, 'Lex Informatica: The Formulation of Information Policy Rules through Technology' (1997) 76 *Tex. L. Rev.* 553. and J. Zittrain, *The Future of the Internet and How to Stop It* (Yale University Press, 2008), p. 107 *et seq*.

74 Perfect in the absolute sense rather than the normative sense. See Lessig (fn 70), preface.

75 K. Yeung, 'Towards an Understanding of Regulation by Design' in Karen Yeung and Roger Brownsword (eds) at *Regulating technologies: Legal futures, regulatory frames and technological fixes* (Hart, 2008), p. 94.

76 Lessig (fn 70), p. 135; R. Brownsword, 'So What Does the World Need Now? Reflections on Regulating Technologies' in Karen Yeung and Roger Brownsword (eds) at Regulating technologies: Legal futures, regulatory frames and technological fixes (Hart, 2008).

77 Yeung (fn 77), pp. 89–90.

78 R. Hoekstra and others, 'The LKIF Core Ontology of Basic Legal Concepts' (2007) 321 *LOAIT*

spectrum: the computer does not understand the normative *substance* of the elements in the ontology, merely the logical relationships between them. Numerous approaches exist to distil legal provisions into pared-down ontologies that are susceptible to logical appraisal, for example the KORA ("Konkretisierung rechtlicher Anforderungen" – concretisation of legal requirements) method,[79] Breaux et al's work on semantic representation of legal norms[80] or Oberle et al's particularly relevant work on formalisation and automated legal reasoning.[81] From the perspective of those designing software systems, however, these approaches require a combination of legal and technical knowledge that only very few systems designers have, decreasing the likelihood that enterprises will embrace them.[82] Later in the discussion of the proposed approach we will see how this problem can be solved by allowing experts in each field to concentrate on what they do best – lawyers focus on the law; software designers focus on the software. The approach is designed to bridge the current gap between those two islands of expertise and make it as easy as possible for enterprises to design software systems which are legally compliant but without the need for significant investment in legal services or regulatory training.

4.3 Shifting the focus from runtime to design

Whereas there are numerous privacy enhancing technologies (PETs) available "off-the-shelf" which can be grafted onto a software system once it has been designed, these are not PbD in the true sense: privacy is not a value represented in *the design* of the software if it is considered as an adjunct only once the design and/or development of the core functionality has been completed. As Hoepman points out,

---

43.

79 V. Hammer, U. Pordesch and A. Roßnagel, 'KORA – Eine Methode Zur Konkretisierung Rechtlicher Anforderungen Zu Technischen Gestaltungsvorschlägen Für Informations-Und Kommunikationssysteme' (1993) 21 *Infotech/I+ G*. (in German. For a useful discussion of KORA in English, see A. Hoffmann and others, 'Towards the Use of Software Requirement Patterns for Legal Requirements' (Social Science Research Network, 2012) SSRN Scholarly Paper ID 2484455, p. 5 *et seq.* <http://papers.ssrn.com/abstract=2484455> accessed 5 July 2015)

80 T.D. Breaux and others, 'Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations' at *Requirements Engineering, 14th IEEE International Conference* (IEEE, 2006) <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1704048> accessed 20 July 2015.

81 D. Oberle and others, 'Engineering Compliant Software: Advising Developers by Automating Legal Reasoning' (2012) 9 *SCRIPTed* 280.

82 P.N. Otto and A.I. Anton, 'Addressing Legal Requirements in Requirements Engineering' at *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International* (2007).

> "[d]uring software development the availability of practical methods to protect privacy is high during actual implementation, but low when starting the project... at the start of the project, during the concept development and analysis phases, the developer stands basically empty handed."[83]

Hafiz provides a useful taxonomy of software techniques for protecting privacy, which he terms "privacy design patterns".[84] Hoepman extends the idea of privacy design patterns into "privacy design strategies", abstracted from the privacy principles contained in the OECD Guidelines,[85] the DPD, and the relevant ISO standard,[86] which can be matched up with the privacy design pattern(s) best suited to implementing them.[87]

4.3.1 Reconceptualising "by design"

As discussed above, an ideal PbD solution should balance the extremes of the policy-architecture spectrum, take account of cost and the state-of-the-art, and be sited at the point in the software development cycle where privacy values can be reflected most efficiently and economically in the design. These various approaches are useful in terms of identifying the appropriate technical mechanism(s) to use to ensure privacy friendliness in a software system, but they are necessarily focused on a point in the design process which *follows* what is arguably the optimum stage for the implementation of PbD. Rather than focussing on the nature of the privacy-enhancing technology itself, we should instead focus on the prior question of whether such a technology is necessary in the first place, with the aim of providing software designers with an early opportunity to side-step the issue altogether by making their systems inherently privacy-friendly.

On this view, we might identify two interpretations of "by design". The orthodox interpretation looks at whether the given product or service has privacy-friendly

83 J.-H. Hoepman, 'Privacy Design Strategies' at *ICT Systems Security and Privacy Protection* (Springer, 2014), p. 1 <http://link.springer.com/chapter/10.1007/978-3-642-55415-5_38> accessed 1 July 2015.

84 M. Hafiz, 'A Collection of Privacy Design Patterns' at *Proceedings of the 2006 conference on Pattern languages of programs* (ACM, 2006); M. Hafiz, 'A Pattern Language for Developing Privacy Enhancing Technologies' (2013) 43 *Software: Practice and Experience* 769.

85 'Guidelines on the Protection of Privacy and Transborder Flows of Personal Data' (fn 8).

86 'ISO/IEC 29100:2011 - Information Technology - Security Techniques - Privacy Framework' (fn 12).

87 Hoepman (fn 84).

features which mitigate or remove what would otherwise be privacy-unfriendly behaviour. Under the alternative perspective, we might consider the software environment within which the product is designed, rather than the actual product *per se*. Instead of burdening the output of the design process with the technological and usability overheads that come with some PETs and other on-the-fly regulatory measures,[88] we instead aim towards the creation of design *environments* where the aims and values of those measures are part of the creative process itself and are subsequently reflected, *by design and by default* (to quote the heading of Article 23(1) of the draft GDPR), in their output. On this view the concept of "by design" might therefore be embodied not in a new privacy-enhancing software approach, but instead in an augmented design process which includes checks to ensure that the end product is *inherently* legally compliant, without the need for technological assessments at runtime. The development environment will help the software designer to answer the question: is this design privacy-friendly? At an abstract level this is akin to Cavoukian's third principle of PbD:

> "[Privacy] is not bolted on as an add-on, after the fact. The result is that privacy becomes an essential component of the core functionality being delivered. Privacy is integral to the system, without diminishing functionality."[89]

References in the PbD literature to the "early design stage"[90] remain too broad; in a software system of any complexity the design stage can be so long that considerations of privacy can be de facto *post hoc* if they happen late enough. In order to avoid this and realise our alternative vision of "by design", there needs to be a means of identifying early on within that stage itself where there is functionality that is potentially hostile to user privacy.

To that end, the method here proposed provides software designers with a tool for modelling parts of their proposed system which can be audited against boilerplate models of the law in order to "certify"[91] that they are legally compliant. As the

88 On the topic of privacy-aware software designers implementing effective but user-unfriendly privacy systems, see A. Whitten and J.D. Tygar, 'Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0' *Usenix Security* (1999).

89 Cavoukian and others (fn 24).

90 See for example European Commission, 'A Digital Agenda for Europe' (fn 17), p. 21.

91 The certification possibilities of the approach are particularly powerful. These are discussed in further detail below.

software model is built out and becomes more complex, these sub-model building blocks can be brought together into a DRM system whose design both enterprises and users can be confident is privacy-friendly *ab initio*.

5. Towards a novel solution: introducing the Petri net

Feigenbaum et al suggest that

> "[a] first step in privacy-aware system design is to analyze the need for information, to graph flows among the various system participants, to analyze how the information flow can be minimized, and to design the message formats accordingly."[92]

The approach introduced here achieves several of these aims simultaneously, and amongst other things promotes the principle of collection limitation which is an important tool in any pragmatic technical implementation of privacy protection.[93]

The Petri net is a well-established, standardised[94] formal modelling approach designed to represent processes in terms of "states" and "transitions". Originally conceived by Carl Petri in 1962,[95] Petri nets have been used in fields as diverse as banking, nuclear power and web services,[96] not to mention for the modelling of legal systems.[97] They are particularly useful in the beginning stages of the design not just of software but of "systems of all kinds in which regulated flows of objects and information are of significance."[98] Despite their graphical appearance and apparent simplicity (they were intentionally designed to facilitate an easy understanding of complex systems[99]), the flow of processes modelled using a Petri net can be easily

92 Feigenbaum and others (fn 34), p. 91.
93 See ibid., pp. 91–92; Spiekermann and Cranor (fn 42); Gürses, Troncoso and Diaz (fn 22).
94 'Systems and Software Engineering. High-Level Petri Nets. Concepts, Definitions and Graphical Notation' (ISO/IEC, 2004) Standard 15909-1:2004+A1:2010.
95 C.A. Petri, 'Kommunikation Mit Automaten' (PhD, University of Bonn 1962) <http://epub.sub.uni-hamburg.de/informatik/volltexte/2011/160/> accessed 10 July 2015.
96 'Systems and Software Engineering. High-Level Petri Nets. Concepts, Definitions and Graphical Notation' (fn 96), p. 1; R. Hamadi and B. Benatallah, 'A Petri Net-Based Model for Web Service Composition' at Proceedings of the 14th Australasian Database Conference - Volume 17 (Australian Computer Society, Inc, 2003); for a comprehensive list with accompanying references see T. Murata, 'Petri Nets: Properties, Analysis and Applications' (1989) 77 Proceedings of the IEEE 541, p. 542.
97 J.A. Meldman, 'A Petri-Net Representation of Civil Procedure' (1977) 19 *Idea* 123; J.A. Meldman and A.W. Holt, 'Petri Nets and Legal Systems' (1971) 12 *Jurimetrics Journal* 65.
98 W. Reisig, *A Primer in Petri Net Design* (Springer, 1992), p. 1.
99 *Ibid.*, p. 2.

simulated[100] and, crucially, can be formally (mathematically) verified. These characteristics mean they can balance intuitive comprehension and analytical certainty in a way which other superficially similar modelling approaches, such as Unified Modelling Language, do not.[101]

Although Meldman's work in the 1970s[102] appears to be the last time in the literature that Petri nets were applied to the modelling of legal processes, the advent of PbD brings with it a renewed impetus to consider approaches which can begin to resolve the conceptual distance between legal regulatory requirements and technical implementation. The particular characteristics of Petri nets mean they can bridge the gap between high-level abstract thinking about processes (what lawyers and managers might do) and low-level, detailed consideration of concrete technical behaviour (what software designers and developers do). As discussed above, the singular lack of overlap in expertise between the legal and technical worlds impedes the kind of collaboration that would lead naturally to the development of approaches which facilitate legal compliance at the software design stage. Petri nets have the potential to mitigate this problem by combining an intuitive method for process modelling that non-technical (legal) experts can understand with a precise and formalised logic that can facilitate the concrete design decisions which technical experts will be required to make in the new world of PbD, particularly in areas such as DRM where privacy concerns are especially acute. Furthermore, the formal proofing characteristics of Petri nets have positive implications for user confidence, and thus marketability, as they facilitate the stricter transparency and accountability requirements of the draft GDPR.

5.1 Petri nets: a primer

5.1.1 Primitives: states, transitions and arcs

The Petri net is a graphical representation of a process, made up of symbols ("primitives") representing *states* and *transitions*.[103] These are connected by *arcs*

100 Using open source tools such as GreatSPN, which was used to draw and validate the models in this paper. See <http://www.di.unito.it/~amparore/mc4cslta/editor.html> accessed 9 August 2015. For a full list of tools, see the list at fn 120, *infra*.

101 K. Salimifard and M. Wright, 'Petri Net-Based Modelling of Workflow Systems: An Overview' (2001) 134 *European Journal of Operational Research* 664, p. 667.

102 Meldman (fn 98).

103 In the literature the term "places" is sometimes used instead of "states". The latter implies a status rather than a physical location, however, so seems more appropriate for our purposes.

(arrows) which represent the flow of the process. These three primitives are the essence of all Petri nets.[104] A simple Petri net is shown in Figure 1.[105]

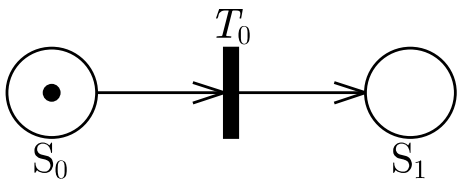

Figure 1

States are represented by circles, while transitions are represented by a rectangle.[106] There are several types of Petri net, each with slightly different characteristics tailored to a particular class of application. For our purposes we are mostly interested in the “timed” Petri net, a variant which allows transitions to be given a duration and prioritised such that they fire in a given chronological order. In such a net the transition symbols can be *immediate* (filled) or *exponential* (unfilled), with any immediate transitions capable of firing doing so before their exponential counterparts.

Returning to the other primitives, a state containing a *marker* (a dot) “holds”, which is to say that that state (or states) represents the current condition of the system within the range of possible conditions contained within the overall process. Multiple states can lead to, or from, a single transition, and they can hold simultaneously. Both of these characteristics are shown in Figure 2. At any given moment the particular configuration of states is called the net's *marking*.

104 Note that this is necessarily a very brief overview of Petri nets and their most basic concepts, as a fuller exposition of the wealth of literature they have generated is outwith the scope of this paper. For a sample, see Petri's original thesis Petri (fn 97) (German) or Murata (fn 97).

105 These figures are inspired by those in Meldman (fn 99) owing to the clarity of the latter.

106 Or sometimes a square, or a line perpendicular to the arrow.

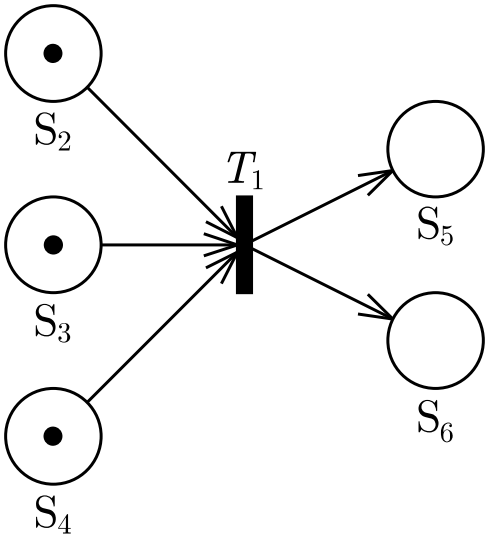

Figure 2

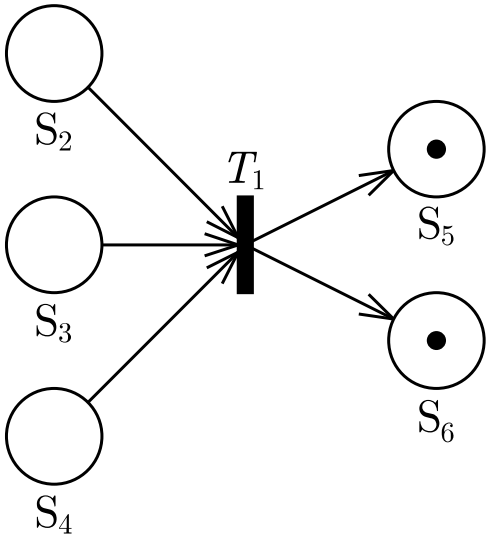

Figure 3

Once a transition fires, all states leading to it will cease holding and those which lead from it will start holding. This is regardless of the number of prior or following states – preceding markers are exhausted and as many new ones as are required to fulfil the following states are created. This is demonstrated in Figures 2 and 3.

5.1.2 Inhibitor arcs and arc weighting

A transition will fire when all the states which lead to it hold. An important exception to this is where the arrow is an *inhibitor arc*. This flips the logic so that the transition is fired when the state from which the inhibitor arc flows does *not* hold. In Figure 4, for example, this would result in S_1 holding.

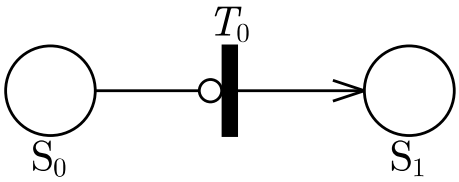

Figure 4

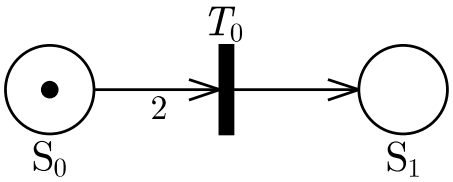

Figure 5

The final basic representation is *arc weighting*, which specifies the minimum number of markers which an arrow will carry – if the arrow has a weighting of more than one, then the transition it points to will fire only if the state which precedes it has at least that number of markers. The weighting is denoted by a number next to the relevant arrow, as in Figure 5 (where the weighting is 1, no number is shown). In Figure 5, then, S_1 cannot hold because there are insufficient markers to fire T_0.

5.1.3 Competing transitions

Two or more transitions can be in competition, or conflict, with one another, as shown in Figure 6. There, because only one of the states necessary to fire T_1 holds, but both required for T_2 hold, the latter will be fired. S_5 will then hold, without the possibility of S_4 holding (Figure 7).

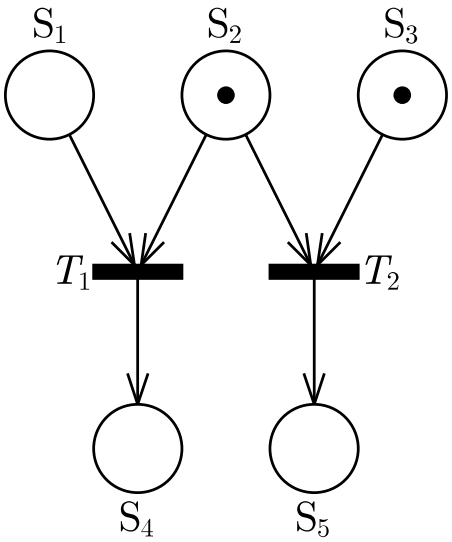

Figure 6

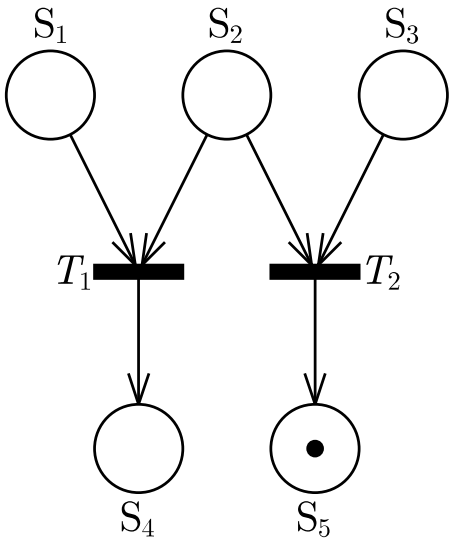

Figure 7

Providing a very limited range of primitives in this way means that complex real-world processes can be reduced to simple logical representations which are suitable for computational simulation and analysis. Indeed, in complex nets the transitions themselves can be represented as "sub-nets", thus mirroring the basic architecture of object-oriented computer programming where discrete methods process input data and pass their output on to other functions (indeed, abstracting and nesting nets in this way is sometimes referred to as "object Petri nets"). For example, Meldmann identifies repeating sub-nets (which he calls "general events") within his complex Petri net of United States civil procedure, which he abstracts into a single "function" that can be called upon when required at various points in the overall net.[107] The detailed model is thus transformed from a complex net containing many tens of primitives to a smaller number of abstracted, high-level sub-nets with comparatively few primitives demonstrating the flow between them. This process of abstraction could theoretically be repeated until ultimately the legal system is modelled in its entirety.

107 Meldman (fn 99), p. 142 et seq. For the complete model see *ibid.,* p. 141.

5.2 Petri nets and PbD

From the perspective of PbD the Petri net is a powerful tool. A newly-proposed software system, and its constituent parts, can be repeatedly tested according to a model of the relevant part(s) of data protection law. These high-level tests can free DRM software designers to concentrate on designing a system that effectively protects the rightholder's intellectual property without having to take on the cognitive load of remembering and applying a complex body of regulatory norms.

From a wider perspective, Petri nets have the benefit of providing “clarification of ambiguities and inconsistencies in the natural language descriptions of systems”.[108] For example, in considering his Petri model of US civil procedure, Meldman notes that a particular outcome which is not readily intelligible from a reading of the legislative provisions alone “comes right to the surface when attempting to describe the rules in the Petri-net language.”[109] The long-term potential then is not just for the creation of a tool that passively aids systems designers and developers but also, through the abstraction and clear presentation of complex legal norms, promotes consideration of those norms in the technical environment, with concomitantly positive implications for user privacy. By going through the modelling process, designers can gain “a detailed understanding of the relevant processes as well as stakeholder needs”,[110] potentially resulting in the ultimate realisation of PbD aims: software designers start to internalise and apply the regulatory norms consciously and naturally, without the external prompting of a tool.

5.2.1 Modelling data protection law

To demonstrate the Petri net in context, this section will model part of the data protection principles from the provisions of the DPD. Those provisions can be thought of essentially as a set of test questions which the controller is expected to ask herself before engaging in processing, such as “are these data personal?”, “do I need to seek explicit consent before processing?”, “is there an exception that permits processing?”, and so on. Each question represents a “gate” which will affect which question(s) must subsequently be asked, and ultimately whether the processing is lawful according to the Directive.

108 Meldman and Holt (fn 99), p. 65.
109 Meldman (fn 99), p. 145.
110 Spiekermann and Cranor (fn 42), p. 69.

Figure 8 shows a Petri (sub-)net which models the DPD Article 8 provision regarding special (sensitive) categories of data. In practice the initial marking of the sub-net would be set according to whether the system being assessed processes one of those special categories of data and whether one of the exceptions applies (these inputs from other parts of the model are represented by dotted arcs). If sensitive data is being processed (in the initial marking one of the states in the top-right box holds), $\mathrm{processing\_can\_continue}$ cannot hold unless one of the exception states in the bottom-right box also holds. The presumption here is that $\mathrm{processing\_is\_legitimate}$ holds as an output of a previous test which checks whether the processing is legitimate pursuant to Article 7. $\mathrm{S_1}$ will hold by default because transition T_1 fires automatically (it is an immediate transition and all of its input states currently hold). Given the initial marking of this sub-net shows one of the special category states holds ($\mathrm{racial\_ethnic\_origin}$), $\mathrm{sensitive\_data\_present}$ will also hold but, because of the absence of an exception state (bottom right box) the transition $\mathrm{article\_8\_exception\_applies}$ can never fire, and thus $\mathrm{processing\_can\_continue}$ is unreachable and the process is "deadlocked". (Note the inhibitor arc between $\mathrm{sensitive\_data\_present}$ and the $\mathrm{non\_sensitive\_data}$ transition – because of its reversing logic, if the former does not hold because sensitive data is not present then the latter will fire, thus bypassing the exception test and moving straight to $\mathrm{processing\_can\_continue}$.)

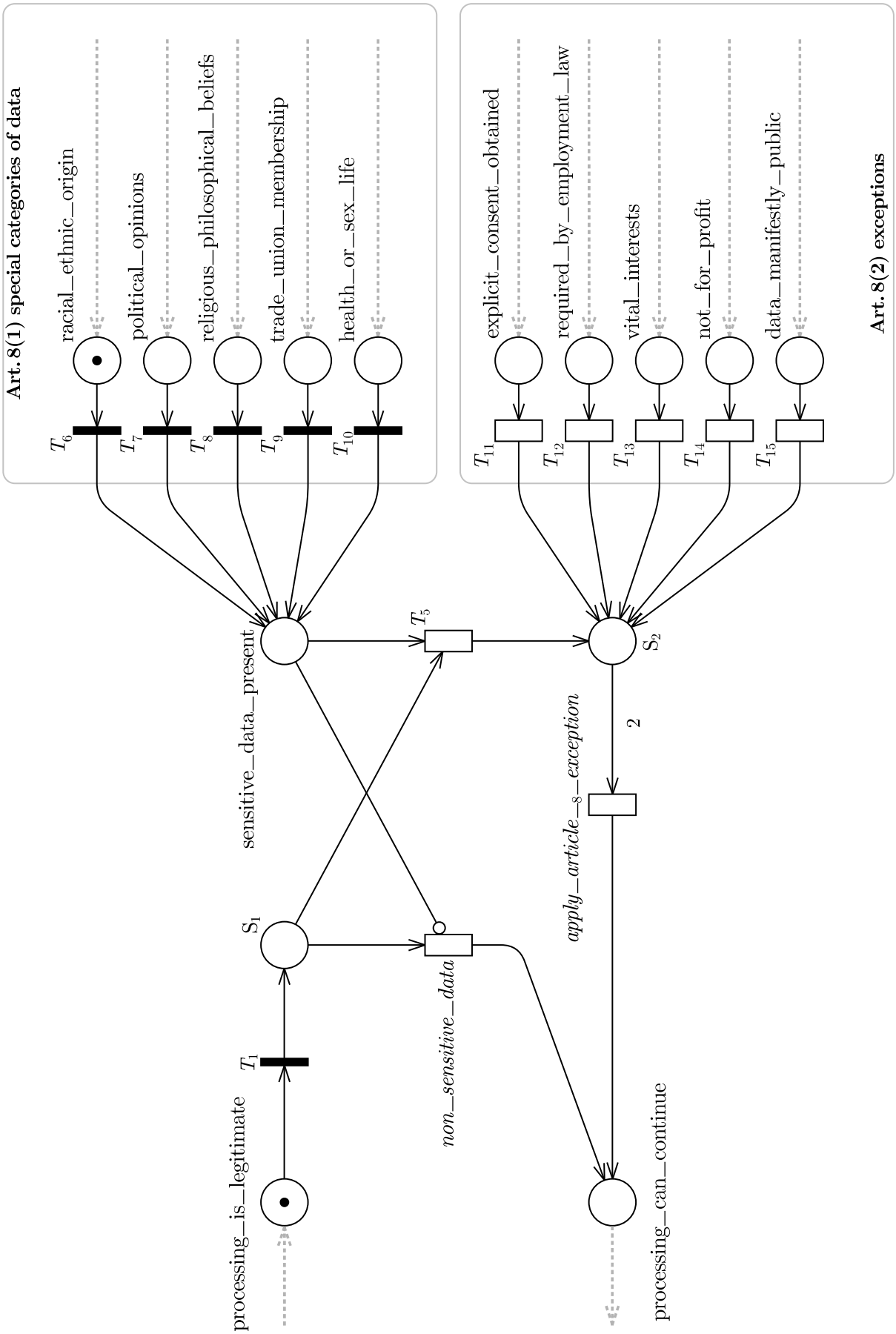
Art. 8(1) special categories of data
racial_ethnic_origin
political_opinions
religious_philosophical_beliefs
trade_union_membership
health_or_sex_life
Art. 8(2) exceptions
explicit_consent_obtained
required_by_employment_law
vital_interests
not_for_profit
data_manifestly_public
processing_is_legitimate
sensitive_data_present
non_sensitive_data
apply_article_8_exception
processing_can_continue
2

Figure 8

5.2.2 Interlinking software and legal models

This is a simple example of one small part of the overall net, and obviously it represents a part of the legal, rather than the software, process. But within this approach models of both can interlink. For example, in Figure 9, in order for the DRM software process to continue from S_0 to S_4, the legal sub-net must be traversed successfully, which includes transition T_2. We can think of that transition as an abstracted version of the special categories of data test represented in Figure 8 above. As we saw there, that test required input regarding the forms of data which were to be processed, and the exceptions which applied. This information can be provided by other parts of the software process, as demonstrated in Figure 9. There we can imagine a sub-net of the overall software net communicating to the legal sub-net (by means of a state being held, or not) whether one of the categories of sensitive data is being processed, and whether any exceptions apply. The information so provided allows the assessment contained in the legal sub-net to be carried out.

A theoretical real-world example will illuminate this concept. The DRM software sub-net in Figure 9 might represent a piece of functionality which presents a form to the user which they must complete in order to access the service represented by the overall software net. One of the optional questions on that form might ask the user to specify their ethnic origin. The software sub-net will therefore pass this state to the legal sub-net depending on whether the user entered that piece of information – if they did, then sensitive information has been gathered, and if not, then it has not (*ceteris paribus*, of course). If we then imagine the Figure 9 net within a model of the wider system, S_1 could represent $\mathrm{processing\_is\_legitimate}$ from Figure 8, S_3 could be $\mathrm{processing\_can\_continue}$, and input from S_2 in the software sub-net becomes $\mathrm{racial\_ethnic\_origin}$.

This is only an abstract and simplified example, of course; in practice the approach would include states and transitions sufficiently detailed to accurately map the inputs and outputs of both the legal and software processes.

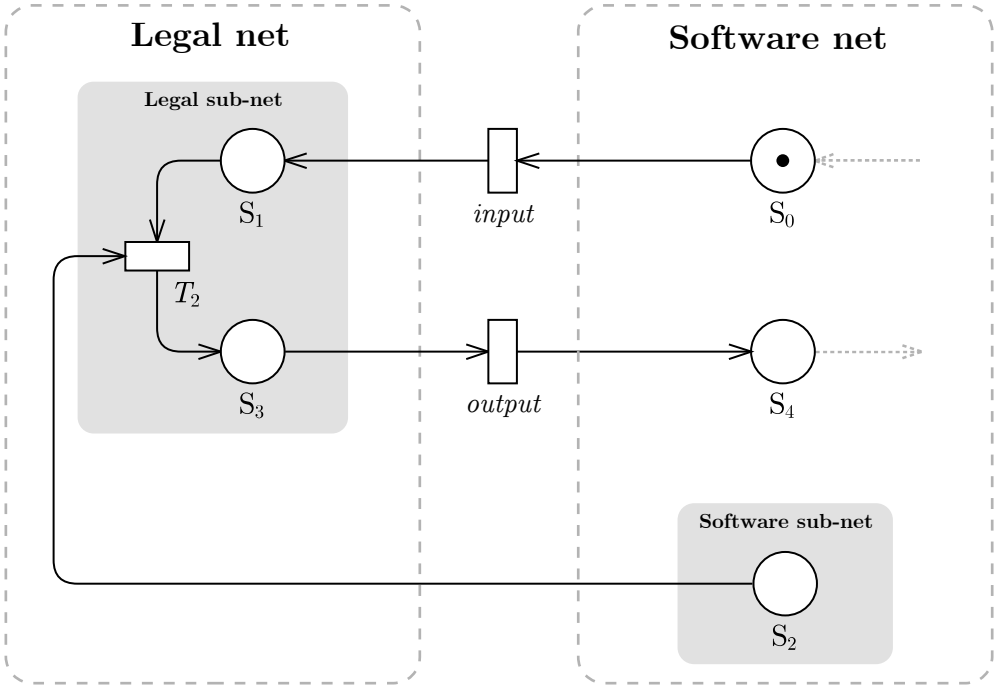

Figure 9

5.3 Validating the model

In software development there are two main approaches to validating an application's behaviour – static and dynamic analysis. The latter is used on "live" code, that is code that has been compiled[111] and is run in a testing environment analogous to that of the end user. The former is like a theoretical test run, where rather than emulating the software's performance in a real-world context with plausible test data, instead it is the underlying logic which is evaluated. The strength of static analysis is that it "can be used in the initial phase of testing to identify definite program errors such as deadlocks that are guaranteed to occur."[112] Petri nets are a tool well suited to this aim because

> "[they allow] specification prototypes to be developed to test ideas at the earliest and cheapest opportunity. Specifications written in the technique may be subjected to analysis methods to prove properties about the specifications, before implementation commences, thus saving on testing and maintenance time and providing a high level of quality assurance."[113]

111 Compilation is the process of turning source code into machine-executable instructions.
112 S.M. Shatz and W.K. Cheng, 'A Petri Net Framework for Automated Static Analysis of Ada Tasking Behavior' (1988) 8 *Journal of Systems and Software* 343, p. 343.
113 'Systems and Software Engineering. High-Level Petri Nets. Concepts, Definitions and Graphical Notation' (fn 96), Introduction.

These attributes are self-evidently attractive from a commercial perspective with respect to complying with PbD in the development of new DRM systems. Shatz and Cheng demonstrate one method for translating software code into a Petri net than can be assessed using appropriate software to identify flow problems.[114] Although their examples are concerned with deadlocks in the Ada programming language, the concepts are readily applicable across various contexts more relevant to our purposes, including more common C-based programming languages,[115] business process modelling,[116] and web services.[117]

As mentioned above, a crucial aspect of the Petri net language is the ability to formally (that is, mathematically) prove the "reachability" of the states within the net. Provided the legal and software systems have been modelled accurately, one can thus certify that the design of a given system is or is not compliant with a particular law. Although a full formal proof is outwith the scope of this paper, there follows a brief discussion which will give a sense of the mathematical underpinnings of the approach.

5.3.1 Formal definition and reachability analysis

Borrowing from Murata's definition,[118] a Petri net is a set of 5 elements ($\mathrm{P} = (\mathrm{S}, \mathrm{T}, \mathrm{F}, \mathrm{W}, \mathrm{M}_0)$) consisting of:

i. $\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3 \ldots, \mathrm{S}_n\}$ (a set of states)

ii. $\mathrm{T} = \{\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3, \ldots, \mathrm{T}_n\}$ (a set of transitions)

iii. $\mathrm{F} \subseteq (\mathrm{S} \times \mathrm{T}) \cup (\mathrm{T} \times \mathrm{S})$ (a set of arcs; the *flow relation*)

iv. $\mathrm{W}\colon \mathrm{F} \rightarrow \{1, 2, 3, \ldots\}$ (the arc weightings)

v. $\mathrm{M}_0\colon \mathrm{S} \rightarrow \{0, 1, 2, \ldots\}$ (the net's initial marking)

vi. $\mathrm{S} \cap \mathrm{T} = \emptyset$ and $\mathrm{S} \cup \mathrm{T} \neq \emptyset$

114 Shatz and Cheng (fn 113).

115 See B. Lin, 'Software Synthesis of Process-Based Concurrent Programs' at *Proceedings of the 35th annual Design Automation Conference* (ACM, 1998) <http://dl.acm.org/citation.cfm?id=277182> accessed 24 July 2015. Most modern software is written in C or in languages based on C.

116 S. Hinz, K. Schmidt and C. Stahl, 'Transforming BPEL to Petri Nets' in Wil MP van der Aalst and others (eds) at *Business Process Management* (Springer Berlin Heidelberg, 2005).

117 Hamadi and Benatallah (fn 97).

118 Murata (fn 98), p. 543.

The “plain” net with no markers is denoted by $\mathrm{N} = (\mathrm{S}, \mathrm{T}, \mathrm{F}, \mathrm{W})$ while a net with a given initial marking is denoted by $(\mathrm{N}, \mathrm{M}_0)$. Taking this forward we can define the (already abstracted) net in Figure 9 as follows:

i. $\mathrm{S} = \{\mathrm{S}_0, \mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3, \mathrm{S}_4\}$
ii. $\mathrm{T} = \{\mathrm{input}, \mathrm{output}, T_2\}$
iii. $\mathrm{F} = \{(\mathrm{S}_0, \mathrm{input}), (\mathrm{input}, \mathrm{S}_1), (\mathrm{S}_1, T_2), (T_2, \mathrm{S}_3), (\mathrm{S}_3, \mathrm{output}), (\mathrm{output}, \mathrm{S}_4), (\mathrm{S}_2, T_2)\}$
iv. $\mathrm{W} = \{(\mathrm{S}_0, \mathrm{input}) \rightarrow 1, (\mathrm{input}, \mathrm{S}_1) \rightarrow 1, (\mathrm{S}_1, T_2) \rightarrow 1, (T_2, \mathrm{S}_3) \rightarrow 1, (\mathrm{S}_3, \mathrm{output}) \rightarrow 1, (\mathrm{output}, \mathrm{S}_4) \rightarrow 1, (\mathrm{S}_2, T_2) \rightarrow 1\}$
v. $\mathrm{M}_0 = \{\mathrm{S}_0 \rightarrow 1\}$

In order to analyse the net to discern its behaviour, we can test its “reachability” to identify whether a given state can hold as the process is stepped through. This allows us to test whether a given (sub-)net can be successfully traversed. To assess reachability first we set out the possible markings of the net by stepping through them, one by one. The order of each marking's set matches the numbering of the states, for example $\{\mathrm{S}_0, \mathrm{S}_1, \mathrm{S}_2, ...\}$, and 0 or 1 denotes the number of markers in that state – any number greater than 0 therefore means that the relevant state holds. For Figure 9 the possible markings are as follows:

$$\mathrm{M}_0 = \{1, 0, 0, 0, 0\}$$
$$\mathrm{M}_1 = \{0, 1, 0, 0, 0\}$$

Since there are no markings in which our desired state S_4 holds, this process is at a deadlock and something needs to change for it to continue. If we alter the net to include an output marker in S_2 in the software sub-net, the possible states look instead like this:

$$\mathrm{M}_0 = \{1, 0, 1, 0, 0\}$$
$$\mathrm{M}_1 = \{0, 1, 1, 0, 0\}$$
$$\mathrm{M}_2 = \{0, 0, 0, 1, 0\}$$
$$\mathrm{M}_3 = \{0, 0, 0, 0, 1\}$$

Under this initial marking it is thus possible for S_4 to hold, and therefore for the software process to continue – it successfully "passes" the test of the legal sub-net.

Clearly the matrix of markings will be larger when derived from a less abstracted net, such as that in Figure 8. There, if for example the system processes data relating to racial or ethnic origin and explicit consent has been obtained (in other words, the initial marking shows $\mathrm{legitimate\_processing}$, $\mathrm{racial\_ethnic\_origin}$ and $\mathrm{explicit\_consent\_obtained}$ holding), then where

$$\begin{aligned}S = \{&\mathrm{legitimate\_processing}, S_1, \mathrm{sensitive\_data}, \mathrm{racial\_ethnic\_origin},\\ &\mathrm{political\_opinions}, \mathrm{religious\_philosophical\_beliefs}, \mathrm{trade\_union\_membership},\\ &\mathrm{health\_or\_sex\_life}, S_2, \mathrm{explicit\_consent\_obtained},\\ &\mathrm{required\_by\_employment\_law}, \mathrm{vital\_interests}, \mathrm{not\_for\_profit},\\ &\mathrm{data\_manifestly\_public}, \mathrm{processing\_can\_continue}\}\end{aligned}$$

we can derive a reachability matrix thus:

$$\begin{aligned}M0 &= \{1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0\}\\ M1 &= \{0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0\}\\ M2 &= \{0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0\}\\ M3 &= \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1\}\end{aligned}$$

which shows that $\mathrm{processing\_can\_continue}$ (the final state in marking M_3) can hold, verifying to the software designer that her system is compliant.

Although the literature on Petri net provability is highly technical (and the introduction shown here necessarily brief), the analytical methods are well-established and are a part of numerous software analysis tools which are already available.[119] If these were integrated into software development environments the underlying mathematical analysis would not need to be exposed to the software designer, who could instead focus on modelling and testing their proposed system.

119 For an exhaustive list, see University of Hamburg, 'Petri Nets World: Petri Nets Tools Database Quick Overview' <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html> accessed 12 August 2015.

5.4 Leveraging the model: triaging deadlocks

Returning to Figure 9, if the software net cannot proceed to S_4 because sensitive data is being gathered and no Article 8(2) exception applies, the system hits a deadlock. Knowing this early on in the DRM development process gives the designer space and time to consider three options:

i. Reassess the overall functionality and aims of the system to determine whether, given the business model being pursued, it would be feasible to achieve the same functionality and commercial outcomes without gathering those data;
ii. Redesign the relevant parts of the system to incorporate technical measures which will make the functionality in question regulatorily compliant; or
iii. Change the business model altogether if its aims are found to be fundamentally incompatible with the requirements of the regulation.

The first option gives the designer (and enterprise) an opportunity to assess their DRM system's proposed behaviour with a view to applying more strictly the collection and use limitation principles of data protection. This accords with both the draft GDPR suggestion of processing minimisation as a potential strategy for achieving PbD,[120] and with the view that technical responses to data protection regulation should focus first on data minimisation, given the technical difficulty at present of implementing more sophisticated forms of automated regulatory compliance.[121]

The second option will usually involve a greater investment of development resources, but when coupled with further rounds of modelling and testing, the incorporation of a PET or privacy design pattern[122] at this early stage of the design process will fulfil Cavoukian's stipulation that “privacy [become] an essential part of the core functionality being delivered”.[123]

The third option will be unattractive to most businesses, particularly startup companies or those on the cusp of developing a disruptive new product or service

120 Recital 61.
121 Gürses, Troncoso and Diaz (fn 21).
122 Hafiz, ‘A Collection of Privacy Design Patterns’ (fn 85).
123 Cavoukian and others (fn 25), Principle 3.

who will not want creative energy and momentum to go to waste. If a product is ultimately unworkable, however, it is clearly preferable to halt its development in the early stages of the design process rather than later on when further creative and financial resources have been invested in it and the risk of waste and loss is accordingly far greater.

5.5 Meeting the GDPR's data controller requirements

Apart from the attraction of automating the detection of deadlocks in the interchange between the software and legal systems, the provability of the proposed approach also makes it possible to fulfil certain assessment requirements of the GDPR. For example Article 22(1) requires controllers to be able to demonstrate that the processing they carry out is “performed in compliance with this Regulation”, Article 28 creates documentation obligations, and Article 33 requires controllers in certain cases to perform a “data protection impact assessment”. In each case the modelling of the system and the results of the reachability analysis can help in meeting these requirements – formal verification can demonstrate that particular states in the model are definitively avoided or reached, while the graphical model itself could serve as documentation of the process being implemented. For the impact assessment, it can also provide evidence of both the behaviour the system was designed to exhibit, and the designer’s efforts to assess it for regulatory compliance.

Related to these requirements are the provisions regarding certification. Recital 60 obliges data controllers to “be able to demonstrate the compliance of processing activities”, while Recital 60(c) refers to “approved certifications”. Article 23(2a) states that “[a]n approved certification mechanism... may be used as an element to demonstrate compliance with the requirements in paragraphs 1 and 2”. Pursuant to Article 39(5), appropriate regulatory bodies (perhaps including the GDPR-proposed European Data Protection Board[124]) could develop, maintain and make available legally sound models of the relevant provisions, using their access to in-house legal expertise and research resources to maintain their accuracy, and perhaps utilising the formalisation techniques discussed above as a means of representing the norms in forms sufficiently abstracted for representation as Petri net states and transitions.

124 See Recital 110. The EDPB is intended to replace the Data Protection Working Party, established under the DPD Art. 29.

Once the legal model has been created and made publicly available, software designers could “plug in” Petri net models of their proposed designs, thus leveraging the regulators’ legal expertise, embodied in the model, without the need to attempt to perform, or commission, an expensive and time-consuming legal assessment. If their proposed system passes the verified legal model this might satisfy one element (if not the whole) of the GDPR’s certification requirements. This in turn would be a selling point for the data controller, demonstrating that they are actively promoting user privacy rather than merely complying with the law, thus strengthening user confidence and driving the market towards PbD compliance. If properly publicised, certifications of this sort would signal that the system in question has been formally verified as privacy-friendly, thus promoting user confidence on an evidential basis in a way that initiatives like TRUSTe, which lack such verification, do not. They would also go some way to fulfilling the data protection principles of openness and accountability discussed above.[125]

6. Concluding remarks

Not surprisingly, the draft GDPR's privacy by design requirements are currently a point of focus for enterprises whose business relies on products and services which process personal data. They are therefore a priority application for the approach set out in this paper. But, as will have been evident, the techniques described need not be limited to data protection law. They are, at least notionally, agnostic with regard to the legal fields they can model and be used to assess, and thus the approach has the potential to assist those designing any software whose functionality has regulatory implications (which is to say almost all of it).

One might see therefore imagine a more holistic approach to implementing compliance in software systems – large bodies of otherwise untested code can be validated against complex, fully-realised models of the legal system, to shed some much-needed regulatory light on what are currently black boxes of functionality that may unwittingly (or not) be compromising user rights or other fundamental social values. By involving the regulatory authorities in a way which simultaneously goes to the heart of the product design but also maintains the freedom to innovate, we retain the democratic connection between legislative intention and the application of the

125 In the GDPR context, see Art. 5(1)(a) and Recital 30.

regulation, without stifling the creative and economic force of the enterprise. Simultaneously, the cultural shift that can take place when enterprises are incentivised to consider regulatory values from the very earliest stages of their design processes (particularly strongly normative values like privacy) can only be good for perceptions of corporate responsibility and therefore, ultimately, economic performance.

Returning to the DRM context, the technology has skewed the commercial relationship grossly in favour of the rightholder, to a point where the user is merely a "second-class participant".[126] Apart from its impact on fundamental rights, the commercial results have also been disastrous, as we have seen. Even if the reputation of DRM is beyond repair, the type of rights protection it was designed to facilitate is necessary, arguably now more than ever. By adopting the approach described here DRM developers can create new generations of content protection that balance the often competing interests of rightholders and users.

126 Vora, Reynolds and Dickinson (fn 37).

Bibliography

Books and journal articles

Acquisti A., Friedman A. and Telang R., 'Is There a Cost to Privacy Breaches? An Event Study' [2006] *ICIS 2006 Proceedings* 94

Breaux T.D. and others, 'Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations' at *Requirements Engineering, 14th IEEE International Conference* (IEEE, 2006) <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1704048> accessed 20 July 2015

Brownsword R., 'So What Does the World Need Now? Reflections on Regulating Technologies' in Karen Yeung and Roger Brownsword (eds) at *Regulating technologies: Legal futures, regulatory frames and technological fixes* (Hart, 2008)

Bygrave L.A., 'Digital Rights Management and Privacy – Legal Aspects in the European Union' in Eberhard Becker, Willms Buhse and Dirk Günnewig (eds) at *Digital Rights Management - Technological, Economic, Legal and Political Aspects* (Springer, 2003)

Cate F.H., 'EU Data Protection Directive, Information Privacy, and the Public Interest, The' (1994) 80 *Iowa L. Rev.* 431

Cavoukian A., 'Privacy by Design: Origins, Meaning, and Prospects for Assuring Privacy and Trust in the Information Era' in George OM Yee (ed) in *Privacy Protection Measures and Technologies in Business Organizations: Aspects and Standards: Aspects and Standards* (IGI Global, 2012)

Cavoukian A. and others, 'Privacy by Design: The 7 Foundational Principles' <https://www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf> accessed 16 May 2015

Chaum D., 'Security Without Identification: Transaction Systems to Make Big Brother Obsolete' (1985) 28 *Commun. ACM* 1030

Cohen J.E., 'A Right to Read Anonymously: A Closer Look at Copyright Management in Cyberspace' (1995) 28 *Conn. L. Rev.* 981

——, 'DRM and Privacy' (2003) 46 *Communications of the ACM* 46

Feigenbaum J. and others, 'Privacy Engineering for Digital Rights Management Systems' in Tomas Sander (ed) at *Security and Privacy in Digital Rights Management* (Springer Berlin Heidelberg, 2002) <http://link.springer.com/chapter/10.1007/3-540-47870-1_6> accessed 16 May 2015

Fuster G.G., *The Emergence of Personal Data Protection as a Fundamental Right of the EU* (Springer Science & Business, 2014)

Gürses S., Troncoso C. and Diaz C., 'Engineering Privacy by Design' (2011) 14 *Computers, Privacy & Data Protection* <https://www.cosic.esat.kuleuven.be/publications/article-1542.pdf> accessed 5 July 2015

Hafiz M., 'A Collection of Privacy Design Patterns' at *Proceedings of the 2006 conference on Pattern languages of programs* (ACM, 2006)

——, 'A Pattern Language for Developing Privacy Enhancing Technologies' (2013) 43 *Software: Practice and Experience* 769

Halderman J.A. and Felten E.W., 'Lessons from the Sony CD DRM Episode.' [2006] *15th USENIX Security Symposium* 77

Hamadi R. and Benatallah B., 'A Petri Net-Based Model for Web Service Composition' at *Proceedings of the 14th Australasian Database Conference - Volume 17* (Australian Computer Society, Inc, 2003)

Hammer V., Pordesch U. and Roßnagel A., 'KORA – Eine Methode Zur Konkretisierung Rechtlicher Anforderungen Zu Technischen Gestaltungsvorschlägen Für Informations-Und Kommunikationssysteme' (1993) 21 *Infotech/I+ G*

Hinz S., Schmidt K. and Stahl C., 'Transforming BPEL to Petri Nets' in Wil MP van der Aalst and others (eds) at *Business Process Management* (Springer Berlin Heidelberg, 2005)

Hoekstra R. and others, 'The LKIF Core Ontology of Basic Legal Concepts' (2007) 321 *LOAIT* 43

Hoepman J.-H., 'Privacy Design Strategies' at *ICT Systems Security and Privacy Protection* (Springer, 2014) <http://link.springer.com/chapter/10.1007/978-3-642-55415-5_38> accessed 1 July 2015

Hoffmann A. and others, 'Towards the Use of Software Requirement Patterns for Legal Requirements' (Social Science Research Network, 2012) SSRN Scholarly Paper ID 2484455 <http://papers.ssrn.com/abstract=2484455> accessed 5 July 2015

Koops B.-J. and Leenes R., 'Privacy Regulation Cannot Be Hardcoded. A Critical Comment on the "privacy by Design" Provision in Data-Protection Law' (2014) 28 *International Review of Law, Computers & Technology* 159

Lessig L., *Code: Version 2.0* (Basic Books, 2006)

Lin B., 'Software Synthesis of Process-Based Concurrent Programs' at *Proceedings of the 35th annual Design Automation Conference* (ACM, 1998) <http://dl.acm.org/citation.cfm?id=277182> accessed 24 July 2015

McPhail T.L., *Global Communication: Theories, Stakeholders, and Trends* (John Wiley & Sons, 2009)

Meldman J.A., 'A Petri-Net Representation of Civil Procedure' (1977) 19 *Idea* 123

Meldman J.A. and Holt A.W., ‘Petri Nets and Legal Systems’ (1971) 12 *Jurimetrics Journal* 65

Mulligan D.K. and Perzanowski A., ‘The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident’ (2007) 22 *Berkeley Technology Law Journal* 1157

Murata T., ‘Petri Nets: Properties, Analysis and Applications’ (1989) 77 *Proceedings of the IEEE* 541

Oberle D. and others, ‘Engineering Compliant Software: Advising Developers by Automating Legal Reasoning’ (2012) 9 *SCRIPTed* 280

Otto P.N. and Anton A.I., ‘Addressing Legal Requirements in Requirements Engineering’ at *Requirements Engineering Conference, 2007. RE ’07. 15th IEEE International* (2007)

Petri C.A., ‘Kommunikation Mit Automaten’ (PhD, University of Bonn 1962) <http://epub.sub.uni-hamburg.de/informatik/volltexte/2011/160/> accessed 10 July 2015

Pocs M., ‘Will the European Commission Be Able to Standardise Legal Technology Design without a Legal Method?’ (2012) 28 *Computer Law & Security Review* 641

Reidenberg J.R., ‘Lex Informatica: The Formulation of Information Policy Rules through Technology’ (1997) 76 *Tex. L. Rev.* 553

Reisig W., *A Primer in Petri Net Design* (Springer, 1992)

Salimifard K. and Wright M., ‘Petri Net-Based Modelling of Workflow Systems: An Overview’ (2001) 134 *European Journal of Operational Research* 664

Schwartz P.M. and Solove D.J., ‘Reconciling Personal Information in the United States and European Union’ (2014) 102 *California Law Review* 877

Shatz S.M. and Cheng W.K., ‘A Petri Net Framework for Automated Static Analysis of Ada Tasking Behavior’ (1988) 8 *Journal of Systems and Software* 343

Spiekermann S. and Cranor L.F., ‘Engineering Privacy’ (2009) 35 *IEEE Transactions on Software Engineering* 67

Whitten A. and Tygar J.D., ‘Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0.’ at *Usenix Security* (1999)

Woodford C., ‘Trusted Computing Or Big Brother - Putting the Rights Back in Digital Rights Management’ (2004) 75 *University of Colorado Law Review* 253

Yeung K., ‘Towards an Understanding of Regulation by Design’ in Karen Yeung and Roger Brownsword (eds) at *Regulating technologies: Legal futures, regulatory frames and technological fixes* (Hart, 2008)

Zittrain J., *The Future of the Internet and How to Stop It* (Yale University Press, 2008)

Institutional publications

Article 29 Data Protection Working Party, ‘Opinion 1/2008 on Data Protection Issues Related to Search Engines’ (2008) WP 148

——, ‘Opinion 01/2012 on the Data Protection Reform Proposals’ (2012) Opinion WP 191 <http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp191_en.pdf> accessed 4 July 2015

European Commission, ‘A Digital Agenda for Europe’ (2010) COM/2010/0245 final

——, ‘A Comprehensive Approach on Personal Data Protection in the European Union’ (2010) COM(2010) 609 final

——, ‘Data Protection Eurobarometer’ (2015) <http://ec.europa.eu/justice/newsroom/data-protection/news/240615_en.htm> accessed 31 July 2015

European Data Protection Supervisor, ‘EDPS Recommendations on the EU’s Options for Data Protection Reform’ (2015) Opinion 3/2015

Federal Communications Commission, ‘AT&T To Pay $25M To Settle Investigation Into Three Data Breaches’ (8 April 2015) <https://www.fcc.gov/document/att-pay-25m-settle-investigation-three-data-breaches-0> accessed 2 August 2015

‘Guidelines on the Protection of Privacy and Transborder Flows of Personal Data’ (OECD, 1980)

‘ISO/IEC 29100:2011 - Information Technology - Security Techniques - Privacy Framework’ (ISO/IEC, 2011) BS ISO/IEC 29100:2011

‘ISO/IEC 15909:2004 - Systems and Software Engineering - High-Level Petri Nets. Concepts, Definitions and Graphical Notation’ (ISO/IEC, 2004) Standard 15909-1:2004+A1:2010

Vora P., Reynolds D. and Dickinson I., ‘Privacy and Digital Rights Management’ at *A position paper for the W3C workshop on Digital Rights Management* (Publishing Systems and Solutions Lab, Hewlett-Packard Laboratories, 2001) <http://www.w3.org/2000/12/drm-ws/pp/hp-poorvi2.html> accessed 16 May 2015

Websites and grey literature

Anderson R., ‘Trusted Computing FAQ v1.1’ (August 2003) <http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html> accessed 15 June 2015

‘Sony BMG Recalls Copy Protected CDs’ (*Billboard*, 18 November 2005) <http://www.billboard.com/articles/news/60609/sony-bmg-recalls-copy-protected-cds> accessed 2 August 2015

Information Commissioner’s Office, ‘Enforcement’ (7 July 2015) <https://ico.org.uk/action-weve-taken/enforcement/> accessed 2 August 2015

In re SONY BMG CD Technologies Litigation settlement agreement [2005] New York Southern District Court No 1:05-CV-09575

Mark Russinovich, ‘More on Sony: Dangerous Decloaking Patch, EULAs and Phoning Home’ <http://blogs.technet.com/b/markrussinovich/archive/2005/11/04/more-on-sony-dangerous-decloaking-patch-eulas-and-phoning-home.aspx> accessed 8 June 2015

‘Viruses, Spyware, and Malware’ (*MIT Information Systems & Technology*) <https://ist.mit.edu/security/malware> accessed 2 August 2015

University of Hamburg, ‘Petri Nets World: Petri Nets Tools Database Quick Overview’ <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html> accessed 12 August 2015